МАТЕМАТИКА

УДК 512.622

А.Э. МАЕВСКИЙ

АЛГОРИТМ ПОИСКА КОРНЕЙ МНОГОЧЛЕНОВ С КОЭФФИЦИЕНТАМИ ИЗ КОЛЬЦА k[x,y]

*Построен детерминированный алгоритм поиска корней многочленов одной переменной с коэффициентами из кольца **k**[x,y], где **k** – произвольное поле. Алгоритм имеет полиномиальные временную и емкостную сложности и может рассматриваться как распространение алгоритма Рота-Рукенштейна [2] поиска корней многочленов с коэффициентами из кольца **k**[x] на случай многочленов с коэффициентами из **k**[x,y].*
Ключевые слова: *корни многочленов, алгоритм Рота-Рукенштейна, факторизация многочленов, линейные делители, конечные поля.*

Введение и постановка задачи. Пусть **k** – поле произвольной характеристики, **k**[x,y] – кольцо многочленов от переменных x, y с коэффициентами из **k**, **k**[x,y][T] ($\cong$**k**[x,y,T]) – кольцо многочленов от переменной T с коэффициентами из **k**[x,y]. Под *полной степенью* deg($f(x,y)$) многочлена $f(x,y)$ ($\in$ **k**[x,y]) будем понимать максимальную из степеней мономов, входящих в $f(x,y)$, а под *степенью* (*T-степенью*) многочлена $Q(x,y,T)$ ($\in$ **k**[x,y][T]) – максимальный показатель степени переменной T, с которым она входит в $Q(x,y,T)$. Многочлен $f(x,y)$ ($\in$ **k**[x,y]) будем называть *T-корнем* многочлена $Q(x,y,T)$, если многочлен $Q(x,y,f(x,y))$ нулевой.

Рассмотрим следующую задачу: для заданного многочлена $Q(x,y,T)$ ($\in$ **k**[x,y][T]) и заданного целого числа d (>0) найти все T-корни $Q(x,y,T)$ полной степени не выше d. Эта задача возникает во многих областях современной математики, например, в теории помехоустойчивого кодирования при решении задач списочного декодирования [1], [2], [5]. Легко показать, что множество

$$\Omega_Q(d) = \{ f(x,y) \in \mathbf{k}[x,y] \mid \deg(f(x,y)) \le d,\ Q(x,y,f(x,y)) \equiv 0 \}$$

всех T-корней $Q(x,y,T)$ полной степени не выше d находится во взаимно однозначном соответствии с множеством делителей $Q(x,y,T)$ вида (T - $f(x,y)$), где deg($f(x,y)$) $\le d$. Поэтому исходная задача эквивалентна задаче поиска всех линейных делителей многочлена $Q(x,y,T)$ вида (T - $f(x,y)$), deg($f(x,y)$) $\le d$.

Существует несколько подходов к решению поставленной задачи. Например, можно использовать общие алгоритмы факторизации многочленов от нескольких переменных [3], [4], и выделить все искомые линейные делители специального вида. Однако вычислительная сложность при этом может оказаться слишком высокой, так как почти все алгоритмы факторизации многочленов от нескольких переменных вероятностные, а многочлен $Q(x,y,T)$ может иметь большое количество ненужных нам линейных делителей вида ($g(x,y)T+f(x,y)$). В работе [5] предложен алгоритм поиска T-корней многочленов с коэффициентами из поля рациональных функций **k**($x_1,\ldots,x_m$). Так как **k**[x_1,x_2]$\subset$**k**($x_1,\ldots,x_m$) при $m \ge 2$, этот алгоритм может быть применен и для построения множества $\Omega_Q(d)$. Однако он использует нетри-

виальную технику алгебраической геометрии и коммутативной алгебры, что сильно затрудняет, с одной стороны, его использование неспециалистами, а с другой, его аппаратную или программную реализацию.

В работе Рота и Рукенштейна [2] построен алгоритм поиска T-корней степени не выше d многочленов с коэффициентами из кольца $\mathbf{k}[x]$. Отметим, что в классе подобных алгоритмов алгоритм из [2] считается одним из самых быстрых и эффективных [5]. По аналогии со схемой построения алгоритма Рота-Рукенштейна в настоящей работе построен алгоритм вычисления $\Omega_Q(d)$, обоснована его корректность и получена оценка его асимптотической сложности.

Алгоритм вычисления множества $\Omega_Q(d)$. Рассмотрим некоторый многочлен $Q(x,y,T)$ из кольца $\mathbf{k}[x,y][\mathrm{T}]$. Определим такое целое неотрицательное число r, что y^r делит $Q(x,y,T)$, но y^{r+1} не делит $Q(x,y,T)$. Положим

$$Q^{(y)}(x,y,T) = Q(x,y,T)/y^r.$$

Пусть $f(x,y) = \sum_{l=0}^{d}\sum_{k=0}^{d-l} f_{kl}x^k y^l$ – некоторый многочлен полной степени d из кольца $\mathbf{k}[x,y]$. Для всех целых $i\in[0,d]$ рекуррентно определим многочлены $f_i(x,y)$ ($\in\mathbf{k}[x,y]$), $Q_i(x,y,T)$ и $Q_i^{(y)}(x,y,T)$ ($\in\mathbf{k}[x,y][T]$) следующим образом. Положим $f_0(x,y) = f(x,y)$, $Q_0(x,y,T) = Q_0^{(y)}(x,y,T) = Q^{(y)}(x,y,T)$,

$$f_i(x,y) = (f_{i-1}(x,y) - f_{i-1}(x,0))/y = \sum_{l=i}^{d}\sum_{k=0}^{d-l} f_{kl}x^k y^{l-i}, \quad (1)$$

$$Q_i(x,y,T) = Q_{i-1}^{(y)}(x,y,yT + f_{i-1}(x,0)), \quad (2)$$

$$Q_i^{(y)}(x,y,T) = Q_i(x,y,T)/y^{r(i)}, \quad (3)$$

где $r(i)$ (≥ 0)– такое целое число, что $y^{r(i)}$ делит $Q_i(x,y,T)$, а $y^{r(i)+1}$ не делит $Q_i(x,y,T)$.

Лемма 1. Рассмотрим произвольное целое число $i\in[1,d]$. Тогда многочлен $(T - f_i(x,y))$ делит многочлен $Q_i^{(y)}(x,y,T)$ в том и только в том случае, когда многочлен $(T - f_{i-1}(x,y))$ делит многочлен $Q_{i-1}^{(y)}(x,y,T)$.

Доказательство. 1($\Rightarrow$). Пусть $(T-f_i(x,y))$ делит многочлен $Q_i^{(y)}(x,y,T)=Q_i(x,y,T)/y^{r(i)}$. Тогда $(T-f_i(x,y))$ делит также $Q_i(x,y,T)=Q_{i-1}^{(y)}(x,y,yT+ +f_{i-1}(x,0))$. Следовательно,

$$Q_{i-1}^{(y)}(x,y,yT + f_{i-1}(x,0)) = (T - f_i(x,y))U(x,y,T)$$

для некоторого $U(x,y,T)$ ($\in\mathbf{k}[x,y][T]$). В последнее равенство подставив вместо T выражение $(T - f_{i-1}(x,0))/y$, получим:

$$Q_{i-1}^{(y)}(x,y,T) = ((T - f_{i-1}(x,0))/y - f_i(x,y))U(x,y,(T - f_{i-1}(x,0))/y).$$

Умножим обе части последнего равенства на y^L, где L достаточно большое натуральное число, и, учитывая (1), получим:

$$y^L Q_{i-1}^{(y)}(x,y,T) = (T - f_{i-1}(x,y))V(x,y,T),\ V(x,y,T)\in\mathbf{k}[x,y][T].$$

Таким образом, $(T - f_{i-1}(x,y))$ делит $y^L Q_{i-1}^{(y)}(x,y,T)$, и так как y^L и $(T - f_{i-1}(x,y))$ взаимно просты, то $(T - f_{i-1}(x,y))$ делит $Q_{i-1}^{(y)}(x,y,T)$.

2($\Leftarrow$). Пусть $(T - f_{i-1}(x,y))$ делит $Q_{i-1}^{(y)}(x,y,T)$, то есть

$$Q_{i-1}^{(y)}(x,y,T) = (T - f_{i-1}(x,y))U(x,y,T)$$

для некоторого многочлена $U(x,y,T)$ ($\in\mathbf{k}[x,y][T]$). Используя это соотношение, из (2) получаем:

$$Q_i(x,y,T) = (yT + f_{i-1}(x,0) - f_{i-1}(x,y))U(x,y,yT + f_{i-1}(x,0)) = y(T - f_i(x,y))V(x,y,T),$$

где $V(x,y,T) = U(x,y,yT + f_{i-1}(x,0))$.

Следовательно, $(T - f_i(x,y))$ делит $Q_i(x,y,T)$, но $Q_i(x,y,T) = y^{r(i)}Q_i^{(y)}(x,y,T)$, поэтому $(T - f_i(x,y))$ делит и $Q_i^{(y)}(x,y,T)$. •

Теорема 2. Следующие утверждения эквивалентны:

(i) $(T - f(x,y))$ делит $Q(x,y,T)$;

(ii) $\exists i \in [1,d]$: $(T - f_i(x,y))$ делит $Q_i^{(y)}(x,y,T)$;

(iii) $\forall i \in [1,d]$: $(T - f_i(x,y))$ делит $Q_i^{(y)}(x,y,T)$;

(iv) T делит $Q_{d+1}(x,y,T)$, где $Q_{d+1}(x,y,T) = Q_d^{(y)}(x,y,yT+f_{0d})$.

Доказательство. (i)⇒(ii). Отметим, что утверждение (i) можно записать как $(T - f_0(x,y))$ делит $Q_0(x,y,T)$. Тогда, согласно лемме 1, $(T - f_1(x,y))$ делит $Q_1(x,y,T)$, следовательно, i в утверждении (ii) можно положить равным 1.

(ii)⇒(iii). Следует из леммы 1.

(iii)⇒(iv). Пусть имеет место (iii). Так как $f_d(x,y)=f_{0d}$, то $Q_d^{(y)}(x,y,T) =(T-f_{0d})U(x,y,T)$. Тогда $Q_{d+1}(x,y,T)=(yT)\ U(x,y,yT + f_{0d})$ и T делит $Q_{d+1}(x,y,T)$.

(iv)⇒(i). Пусть T делит $Q_{d+1}(x,y,T) = Q_d^{(y)}(x,y,yT+f_{0d})$. Используя те же рассуждения, что и при доказательстве первой части леммы 1, и тот факт, что $f_d(x,y) = f_{0d}$, получаем, что $(T - f_d(x,y))$ делит $Q_d(x,y,T)$. Используя далее лемму 1, получаем, что $(T - f(x,y))$ делит $Q(x,y,T)$. •

Для всех целых $i\in[0,d]$ рекуррентно определим многочлены $h_i(x)$ ($\in$ **k**$[x]$), $M_i(x,T)$ ($\in$ **k**$[x][T]$) следующим образом:

$$h_i(x) = f_i(x,0) = \sum_{k=0}^{d-i} f_{ki}x^k\ , \tag{4}$$

$$M_i(x,T) = Q_i^{(y)}(x,0,T). \tag{5}$$

Отметим, что если многочлен $Q_i^{(y)}(x,y,T)$ ненулевой, то таким же будет и $M_i(x,T)$.

Лемма 3. Если $(T - f(x,y))$ делит $Q(x,y,T)$, то для всех целых $i\in[0,d]$ многочлен $(T - h_i(x))$ делит многочлен $M_i(x,T)$.

Доказательство. Пусть $(T - f(x,y))$ делит $Q(x,y,T)$. Тогда, согласно лемме 1, многочлен $(T - f_i(x,y))$ делит $Q_i^{(y)}(x,y,T)$ для всех целых $i\in[0,d]$. Подставив y=0 в $(T - f_i(x,y))$ и $Q_i^{(y)}(x,y,T)$, получим утверждение леммы. •

Из последней леммы вытекает следующая важная теорема.

Теорема 4. Если $(T - f(x,y))$ делит $Q(x,y,T)$, то для всех целых $i\in[0,d]$ коэффициенты $f_{0i},\ldots,f_{(d-i)i}$ многочлена $f(x,y)$ совпадают с коэффициентами $h_0,\ldots,h_{d-i}$ одного из делителей многочлена $M_i(x,T)$ вида $(T - h(x))$, $\deg(h(x)) = d-i$.•

Последняя теорема доставляет нам способ нахождения множества $\Omega_Q(d)$. По заданному многочлену $Q(x,y,T)$ вычислим многочлен $M_0(x,T)=Q_0^{(y)}(x,0,T)$. Используя, например, алгоритм Рота-Рукенштейна [2], найдем все его T-корни степени не выше d. Согласно теореме 4, коэффициенты $f_{00},\ldots,f_{d0}$ любого T-корня $f(x,y)$ многочлена $Q(x,y,T)$ обязательно совпадают с коэффициентами $h_0,\ldots,h_d$ какого-либо T-корня многочлена $M_0(x,T)$. Далее, для каждого полученного набора коэффициентов $h_0,\ldots,h_d$ согласно (2) и (3), вычислим многочлены $Q_1^{(y)}(x,y,T)$ и $M_1(x,T)$. Определив все T-корни $M_1(x,T)$ полной степени d-1, получим варианты значений коэффициентов $f_{01},\ldots,f_{d1}$. Продолжая процесс поиска коэффициентов многочлена $f(x,y)$, мы, как будет показано в теореме 5, найдем $\Omega_Q(d)$.

Описанный выше процесс формализуется в виде рекурсивного алгоритма FindRoots. В ходе своей работы алгоритм использует две дополнительные переменные: неотрицательное целое число i, определяющее уровень (глубину) рекурсии, и многочлен $f(x,y)$, который изменяется на каждом уровне рекурсии, а на последнем уровне становится кандидатом

на T-корень. При начальном вызове алгоритма FindRoots оба параметра i и $f(x,y)$ должны быть нулевыми.

Алгоритм FindRoots (поиск для многочлена $Q(x,y,T)$ всех T-корней полной степени не выше d):

Вход: многочлен $Q(x,y,T)$ ($\in \mathbf{k}[x,y][T]$), натуральное число d, неотрицательное целое число i, многочлен $f(x,y)$ ($\in \mathbf{k}[x,y]$);

Выход: множество T-корней полной степени d многочлена $Q(x,y,T)$.

Ш1. Найти такое целое неотрицательное число r, что y^r делит $Q(x,y,T)$, но y^{r+1} не делит $Q(x,y,T)$;

Ш2. Положить $Q^{(y)}(x,y,T) := Q(x,y,T)/y^r$;

Ш3. Для каждого из T-корней $h_k(x)$ степени d-i многочлена $Q^{(y)}(x,0,T)$ выполнить:

Ш3.1. Положить $\tilde{R}(x,y,T) := Q^{(y)}(x,y,T+h_k(x))$;

Ш3.2. Положить $R(x,y,T) := \tilde{R}(x,y,yT)$;

Ш3.3. Если ($i = d$), то

Ш3.3.1. Если (T делит $R(x,y,T)$), то вернуть $f(x,y) + y^i h_k(x)$ и выйти из алгоритма, иначе

Ш3.3.2. Выйти из алгоритма;

Ш3.4. Выполнить **FindRoots**($R(x,y,T)$, d, i+1, $f(x,y) + y^i h_k(x)$).

Конец алгоритма.

Теорема 5. Алгоритм FindRoots, запущенный с начальными параметрами ($Q(x,y,T)$,d,0,0), возвращает все T-корни многочлена $Q(x,y,T)$ степени не выше d.

Доказательство. Пусть $\Theta_Q(d)$ ($\subset \mathbf{k}[x,y]$) – множество всех многочленов, возвращаемых алгоритмом FindRoots ($Q(x,y,T)$,d,0,0). Покажем, что $\Theta_Q(d) = \Omega_Q(d)$. Пусть $f(x,y) \in \Theta_Q(d)$. Легко проверить, что полная степень $f(x,y)$ не превышает d. Многочлен $f(x,y)$ построен таким образом, что T делит $R(x,y,T)=Q_d^{(y)}(x,y,yT+f_{0d})$. Поэтому, согласно теореме 2, $f(x,y)$–T-корень $Q(x,y,T)$ и $\Theta_Q(d) \subset \Omega_Q(d)$. Включение $\Omega_Q(d) \subset \Theta_Q(d)$ вытекает из теоремы 4 и того факта, что алгоритм FindRoots на каждом уровне рекурсии i осуществляет полный перебор делителей вида (T - $h_k(x)$) многочлена $Q^{(y)}(x,0,T)$.•

Анализ сложности алгоритма FindRoots. На каждом уровне рекурсии i алгоритма FindRoots мы находим T-корни некоторого ненулевого многочлена, и для каждого из его корней переходим на следующий уровень рекурсии. Может показаться, что совокупное количество корней при переходе с одного уровня рекурсии на другой растет экспоненциально, однако, рассматриваемая ниже лемма 6 показывает, что это не так.

Лемма 6. Пусть $Q(x,y,T) = \sum_{k=0}^{b} q_k(x,y)T^k$ ($\in \mathbf{k}[x,y][T]$) – такой ненулевой многочлен T-степени b, что y не делит $Q(x,y,T)$, $h(x)$ ($\in \mathbf{k}[x]$)–T-корень многочлена $Q(x,0,T)$ степени не выше d и кратности γ. Положим $P_y(x,y,T)=Q(x,y,yT+h(x))$, $P(x,y,T) = P_y(x,y,T)/y^r$, где r – такое наибольшее неотрицательное целое число, что y^r делит $P_y(x,y,T)$, а y^{r+1} не делит $P_y(x,y,T)$. Тогда T-степень многочлена $M(x,T) = P(x,0,T)$ не превосходит γ.

Доказательство. Пусть T-степень $Q(x,y,T)$ равна b,

$$S(x,y,T) = Q(x,y,T + h(x)) = \sum_{k=0}^{b} s_k(x,y)T^k\,,$$

$$P_y(x,y,T) = Q(x,y,yT + h(x)) = \sum_{k=0}^{b} s_k(x,y)y^k T^k\,.$$

Поскольку $h(x)$ является T-корнем кратности γ многочлена $Q(x,0,T)$, то $Q(x,0,T)=(T - h(x))^{\gamma}U(x,T)$ для некоторого $U(x,T)$ ($\in \mathbf{k}[x][T]$), и $S(x,0,T)=Q(x,0,T)= (T)^{\gamma}U(x,T+h(x))$. Так как T^{γ} делит $S(x,0,T)$, то $s_k(x,0) = 0$ при $k\in[0,\gamma\text{-}1]$, но $s_{\gamma+1}(x,0)\neq 0$. Последнее равнозначно тому, что y делит многочлены $s_k(x,y)$ при $k\in[0,\gamma\text{-}1]$ и не делит $s_{\gamma+1}(x,0)$ ($\neq 0$). Следовательно, y делит $P_y(x,y,T)$, но $y^{\gamma+1}$ не делит $P_y(x,y,T)$. Таким образом, $r \in [1,\gamma]$. Запишем многочлен $P(x,y,T)$:

$$P(x,y,T) = P_y(x,y,T)/y^r = \sum_{k=0}^{r} s_k(x,y)y^kT^k \ / y^r + \sum_{k=r+1}^{b} s_k(x,y)y^{k-r}T^k .$$

Теперь видно, что T-степень $M(x,T)=P(x,0,T)$ не превышает r ($\in[1,\gamma]$). •

Следствие. Рассмотрим случай, когда на вход алгоритма FindRoots поступает такой многочлен $Q(x,y,T)$, что все T-корни многочлена $Q^{(y)}(x,0,T)$ имеют кратность 1. Тогда многочлены $Q^{(y)}(x,0,T)$, получаемые на всех последующих уровнях рекурсии, будут иметь T-степень не выше 1, и их T-корни могут быть вычислены непосредственно. •

Следующие две леммы являются подготовительными для теоремы об оценке сложности алгоритма FindRoots.

Лемма 7. Пусть $Q(x,y,T)$ ($\in \mathbf{k}[x,y][T]$) – ненулевой многочлен T-степени b. Тогда количество многочленов, возвращаемых алгоритмом FindRoots, вызванным с параметрами ($Q(x,y,T),d,0,0$), не превышает b, а общее количество рекурсивных обращений алгоритма FindRoots самому к себе не превышает bd.

Доказательство. Для каждого уровня рекурсии i ($\in[0,d]$) алгоритма FindRoots через ω_i обозначим сумму T-степеней всех многочленов $Q^{(y)}(x,0,T)$, возникающих на шаге *Ш3*. Другими словами, ω_i равняется сумме всех T-степеней многочленов $M_i(x,T)$, возникающих в процессе поиска T-корней $Q(x,y,T)$. При $i=0$ существует единственный многочлен $M_0(x,T)=Q^{(y)}(x,0,T)$ T-степени не выше b, поэтому $\omega_0 \le b$. Согласно лемме 6, имеет место неравенство $\omega_i \le \omega_{i-1}$ для каждого $i \in [1,d]$. Следовательно, $\omega_i \le b$ для каждого $i \in [0,d]$. В итоге алгоритм FindRoots при $i = d$ работает не более чем с $\omega_d \le b$ многочленами, и общее количество рекурсивных вызовов FindRoots не превышает $\sum_{i=0}^{d-1}\omega_i \le bd$. •

Лемма 8. Пусть $Q(x,y,T) = \sum_{k=0}^{b} q_k(x,y)T^k$ ($\in \mathbf{k}[x,y][T]$) – такой ненулевой многочлен T-степени b, что полная степень любого коэффициента $q_k(x,y)$ не превосходит m. Тогда T-степень всех многочленов на любом из уровней рекурсии алгоритма FindRoots не выше b, а полная степень коэффициентов этих многочленов на уровне рекурсии i не превышает $O(m+bd^2)$.

Доказательство. Нетрудно видеть, что шаги *Ш1*, *Ш2*, *Ш3.1*, *Ш3.2* не увеличивают T-степень. Пусть

$$\tilde{R}_i(x,y,T) = Q_i^{(y)}(x,y,T + h_i(x)) = \sum_{k=0}^{b} q_{ki}^{(y)}(x,y)(T + h_i(x))^k = \sum_{k=0}^{b} \tilde{r}_{ki}(x,y)T^k ,$$

$$R_i(x,y,T) = \tilde{R}_i(x,y,yT) = \sum_{k=0}^{b} \tilde{r}_{ki}(x,y)y^kT^k$$

– многочлены, вычисляемые на шагах *Ш3.1*, *Ш3.2* алгоритма FindRoots, находящегося на i-м уровне рекурсии, и $\deg(q_{ki}^{(y)}(x,y)) \le m(i)$. Тогда $\deg(\tilde{r}_{ki}(x,y)) \le m(i)+b\deg(h_i(x))=m(i)+b(d-i)$, а $\deg(y^k\tilde{r}_{ki}(x,y)) \le m(i)+b(d-i+1)$. Так

как $m(0)=m$, то $\deg(y^k\tilde{r}_{ki}(x,y))\le m+b(i+1)(d+1-i/2)\le m+b(d+1)(d/2+1)= = O(m + bd^2)$.•

Оценим асимптотическую сложность алгоритма FindRoots. Выберем следующую широко распространенную в теории многочленов модель вычислений [2-4]. Предположим, что базовые операции поля **k** (сложение, вычитание, умножение, деление элементов), а также операции сравнения и присваивания имеют временную сложность $O(1)$. Для многочленов $g(x)$, $h(x)$ ($\in$ **k**$[x]$) операции сложения и вычитания имеют временную сложность $O(\min\{\deg(g(x)), \deg(h(x))\})$ операций поля **k**, а умножение имеет временную сложность $O(\deg(g(x))\cdot\deg(h(x)))$ операций поля **k**.

Теорема 9. Пусть многочлен $Q(x,y,T)$ удовлетворяет условиям леммы 8. Тогда алгоритм FindRoots, вызванный с параметрами ($Q(x,y,T)$, d, 0, 0), имеет временную сложность $O(b^2d^2(b(m+bd^2)^2 + F(b)))$ операций поля **k**, где $F(b)$ – временная сложность алгоритма поиска корней многочлена $Q(T)$ ($\in$ **k**$[T]$) степени b, и емкостную сложность $O(b(m+bd^2)^2)$ элементов поля **k.**

Доказательство. На шагах *Ш1*, *Ш2*, *Ш3.2* изменения затрагивают только мономы, содержащие y. Если коэффициенты $q_k(x,y)$ многочлена $Q(x,y,T)$ записать как элементы **k**$[x][y]$, то, согласно лемме 8, общее количество изменяемых мономов можно оценить как $O(b(m+bd^2))$. Так как в течение работы всего алгоритма шаги *Ш1*, *Ш2*, *Ш3.2* выполняются не более, чем bd раз (лемма 7), то общий вклад этих шагов во временную сложность алгоритма равен $O(b^2d(m+bd^2))$.

Шаг *Ш3.1* удобно выполнять с помощью схемы Горнера. Нетрудно проверить, что в этом случае он имеет временную сложность $O(b^2d(m+bd^2)^2)$. Так как он выполняется в алгоритме bd раз, его вклад в общую временную сложность алгоритма составляет $O(b^3d^2(m+bd^2)^2)$.

Для поиска всех T-корней многочлена $Q^{(y)}(x,0,T)$ на шаге *Ш3* можно воспользоваться алгоритмом Рота-Рукенштейна [2]. Его временную сложность в зависимости от b, d и m можно оценить как $O(bd(b^2(m+bd)+F(b)))$, где $F(b)$ – временная сложность алгоритма поиска корней многочлена $Q(T)$ ($\in$ **k**$[T]$) степени b. Следовательно, в течение работы всего алгоритма шаг *Ш3* имеет сложность $O(b^2d^2(b^2(m+bd)+F(b)))$. Таким образом, общая временная сложность алгоритма FindRoots составляет $O(b^2d^2(b(m+bd^2)^2+F(b)))$ операций в поле **k**.

В алгоритме FindRoots больше всего памяти требуется для хранения на каждом уровне рекурсии коэффициентов многочленов $Q(x,y,T)$. Поэтому оценка $O(b(m+bd^2)^2)$ на емкостную сложность алгоритма следует из того, что общее число мономов $Q(x,y,T)$ не превышает $O((m + bd^2)^2)$, а общее число хранимых многочленов не выше b (по числу параллельно вычисляемых T-корней). •

Необходимо отметить, что оценку сложности алгоритма FindRoots можно улучшить, с одной стороны, используя быстрые методы вычислений с многочленами, а с другой, более точным подсчетом числа операций.

Выводы. В работе построен детерминированный алгоритм поиска всех T-корней степени не выше d произвольного многочлена $Q(x,y,T)$ ($\in$ **k**$[x,y][T]$) и доказана его корректность. Показано, что алгоритм имеет полиномиальные временную сложность $O(b^2d^2(b(m+bd^2)^2 + F(b)))$ операций

поля **k** и емкостную сложность $O(b(m+bd^2)^2)$ элементов поля **k**. Алгоритм может быть применен при решении различных задач, например, задачи списочного декодирования некоторых семейств алгебро-геометри-ческих кодов [1].

Библиографический список

1. Маевский А.Э. О списочном декодировании одного класса алгебро-геометрических кодов на проективных кривых // Тр. участников международ. школы-семинара по геометрии и анализу памяти Н.В.Ефимова. – Абрау-Дюрсо, 5-11 сентября, 2006. – Ростов н/Д, 2006. – С. 55-56.

2. Roth R.M., Ruckenstein G. Efficient decoding of Reed-Solomon codes beyond half the minimum distance // IEEE Transactions on Information Theory. – Vol. 46, no. 1, January 2000. – P. 246-257.

3. Gathen J., Kaltofen E. Polynomal-time factorization of multivariate polynomials over finite fields // Lecture Notes in Computer Science. Springer-Verlag. – Vol. 154, 1983. – P. 250-262.

4. Shoup V. A computational introduction to number theory and algebra. – N.-Y.: Cambridge University Press, 2005. – 534 p.

5. Wu X.W., Siegel P.H. Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes // IEEE Transactions on Information Theory. – Vol. 47, no. 6, September 2001. – P. 2579-2587.

Материал поступил в редакцию 16.11.06.

A.E. MAEVSKIY

ROOT-FINDING ALGORITHM FOR UNIVARIATE POLYNOMIALS WITH COEFFICIENTS FROM k[*x,y*]

Deterministic polynomial-time root-finding algorithm for univariate polynomials with coefficients from **k**[x,y] where **k** is a field of any characteristic is constructed. Our algorithm can be viewed as an extension of the Roth-Ruckenstein's root-finding algorithm to polynomials from **k**[x,y][T].

МАЕВСКИЙ Алексей Эдуардович (р.1981), аспирант кафедры «Программное обеспечение вычислительной техники и автоматизированных систем» ДГТУ. Окончил ДГТУ (2003).
Сфера научных интересов: теория помехоустойчивого кодирования, алгебраическая геометрия и коммутативная алгебра, математические методы в защите информации.
Автор 7 научных работ.